1. (10 points) What is the distance between two parallel hyperplanes $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^\top \boldsymbol{x} = b_1\}$ and $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^\top \boldsymbol{x} = b_2\}$? *Hint.* Let $\boldsymbol{a}^\top \boldsymbol{x}_1 = b_1$, $\boldsymbol{a}^\top \boldsymbol{x}_2 = b_2$, and minimize $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2$.

Ans:) = Hyperplanes are given as =

$$\text{hyperplane 1} = \{x \in R^n \mid a^T x = b_1\}$$

$$\text{hyperplane 2} = \{x \in R^n \mid a^T x = b_2\}$$

let x_1 be any point in the first hyperplane and consider that there is a line 'l' which is passing through x_1 in the direction of the normal vector a.

Since point x_1 is on hyperplane and line l is passing through it we can write the equation of line l as =

$$l = x_1 + at, \quad \forall\ t \in R$$

the line is intersecting the hyperplane 2. Hence intersection of hyperplane 2 with line l is given as =

$$a^T(x_1 + at) = b_2$$

$$\Rightarrow a^T x_1 + a^T a t = b_2$$

$$\Rightarrow a^T a t = b_2 - a^T x_1$$

$$\Rightarrow t = \frac{b_2 - a^T x_1}{a^T a}$$

But we know that $\boxed{a^T x_1 = b_1}$ Hence we can write t as =

$$\boxed{t = \frac{b_2 - b_1}{a^T a}}$$

Therefore the intersection point on Hyperplane 2 will be given as =

$$\boxed{x_2 = x_1 + a\frac{(b_2 - b_1)}{a^T a}}$$

The distance between the points x_1, x_2 is the distance between the hyperplanes. from above equation we can obtain =

$$x_2 - x_1 = a \frac{(b_2 - b_1)}{a^T a}$$

taking norm to find distance =

$$\|x_1 - x_2\| = \frac{\|a\| |(b_2 - b_1)|}{a^T a}$$

$$\boxed{\|x_1 - x_2\| = \frac{|b_2 - b_1|}{\|a\|}}$$

2. (20 points) Let x be a real-valued random variable with sample space $\{a_1, \ldots, a_k\}$ where $a_1 \leq a_2 \leq \cdots \leq a_k$. This can be view as a categorical random variable with each catogory assigned a real value. Let $\Pr[x = a_i] = p_i$, then the vector $\boldsymbol{p}$ satisfies $\boldsymbol{p} \geq 0$ and $\boldsymbol{1}^\top \boldsymbol{p} = 1$, i.e., it lies in the probability simplex Δ. For each of the following functions of $\boldsymbol{p}$ on the probability simplex, determine if the function is convex, concave, or neither.

   (a) $\mathrm{E}[x]$
   (b) $\Pr[x > \alpha]$
   (c) $\Pr[\alpha < x < \beta]$
   (d) $-\sum_{i=1}^{k} p_i \log p_i$, the entropy of this distribution
   (e) $\mathrm{var}(x)$

Ans-2 = x is a real valued random variable with sample space $\{a_1, a_2, \ldots, a_k\}$. where $(a_1 \leq a_2 \leq \cdots \leq a_k)$.

Also given that $\Pr[x = a_i] = p_i$
then the vector p satisfies $p \geq 0$ and $1^T p = 1$

(Part a) =

$$E(x) = a_1 p_1 + a_2 p_2 + \cdots + a_k p_k = a^T p$$

This function is a linear function.
Hence it is both concave & convex.

(Part b) =

$$Pr[x > \alpha] = P_{\alpha+1} + P_{\alpha+2} + \cdots + P_K$$

$$= L^T P$$

Here L starts from $(\alpha+1)$ to K and all terms before $(\alpha+1)$ are 0.

again this is also a linear function Hence it is both convex and concave.

(Part c) =

$$Pr[\alpha < x < \beta] = P_{\alpha+1} + P_{\alpha+2} \cdots P_\beta$$

$$= L^T P$$

Just like (Part b), Here also we can interpret $Pr[\alpha < x < \beta]$ as $L^T P$.

where $L[0:\alpha] = 0$

$L[\alpha+1 : \beta] = 1$

$L[\beta : K] = 0$

Since this function is also linear function Hence it is also both concave & convex.

(Part d) $= -\sum_{i=1}^{k} p_i \log p_i$

we know that $\log x$ is concave function.

But $x \log x$ is a convex function.

Hence $\sum_{i=1}^{k} p_i \log p_i$ is also a convex function.

But we know that →

if $f(x) \longrightarrow$ convex

Then $-f(x) \longrightarrow$ concave

Hence $-\sum_{i=1}^{k} p_i \log p_i$ is a concave function

(Part e) =

we know that, (variance) ⇒

$$\text{var}(x) = E_p[x^2] - E_p[x]^2$$

also for $\lambda \in [0,1]$ we can write that =

$$E_{\lambda p + (1-\lambda)q}[x^2] = \lambda E_p[x^2] + (1-\lambda) E_q[x^2]$$

x^2 is a convex function also the Jensen Inequality gives us =

$$\lambda E_p[x]^2 + (1-\lambda) E_q[x]^2 \geq \left(\lambda E_p[x] + (1-\lambda) E_q[x]\right)^2$$

$$= E_{\lambda p + (1-\lambda)q}[x]^2$$

Thus we can infer =

$$\text{var}_{\lambda p + (1-\lambda)q}(x) = E_{\lambda p + (1-\lambda)q}[x^2] - E_{\lambda p + (1-\lambda)q}[x]^2$$

$$\geq \lambda \text{var}_p(x) + (1-\lambda)\text{var}_q(x)$$

This clearly indicates that variance $\text{Var}(x)$ is concave.

3. (10 points) Show that the following two convex problems are equivalent. Carefully explain how the solution of (b) is obtained from the solution of (a).

(a) The robust least squares problem

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \sum_{i=1}^{n} h(\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - \psi_i),$$

where $h : \mathbb{R} \to \mathbb{R}$ is the Huber function defined (with a constant M) as

$$h(t) = \begin{cases} t^2 & |t| \le M \\ M(2|t| - M) & |t| > M. \end{cases}$$

(b) The quadratic program

$$\begin{array}{ll} \underset{\boldsymbol{\theta},\boldsymbol{u},\boldsymbol{v}}{\text{minimize}} & \sum_{i=1}^{n} (u_i^2 + 2Mv_i) \\ \text{subject to} & -\boldsymbol{u} - \boldsymbol{v} \le \boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{\psi} \le \boldsymbol{u} + \boldsymbol{v} \\ & 0 \le \boldsymbol{u} \le M\boldsymbol{1}, \qquad \boldsymbol{v} \ge 0. \end{array}$$

Ans-3 = we have got,

robust least square problem ⇒

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{n} h(\phi_i^T \theta - \psi_i)$$

where $h(t)$ is huber function

and the quadratic program =

$$\underset{\theta,u,v}{\text{minimize}} \sum_{i=1}^{n} (u_i^2 + 2Mv_i)$$

which is subject to $-u - v \le \phi^T \theta - \psi \le u + v$

and $0 \le u \le M1$, $v \ge 0$.

we are given that $\phi^T\theta - \psi \leq u + v$

we can rewrite it as =

$$|\phi_i^T\theta - \psi_i| \leq u_i + v_i$$

where $u_i, v_i > 0$, and u_i and v_i has to be minimized.

To minimize v_i we can write v_i as =

$$\boxed{v_i = |\phi_i^T\theta - \psi_i) - u_i} \longrightarrow (1)$$

lets substitute this value to the Quadratic program =

$$\underset{\theta, u}{\text{minimize}} \sum_{i=1}^{n} \left(u_i^2 + 2M\left(|\phi_i^T\theta - \psi_i| - u_i\right)\right)$$

$$= \sum_{i=1}^{n} \left(u_i^2 + 2M|L_i| - 2Mu_i\right) \longrightarrow (2)$$

if we assume that $L_i = |\phi_i^T\theta - \psi_i|$

Then we have $u_i = \begin{cases} |L_i| & \text{when } |L_i| \leq M \\ M & |L_i| > M \end{cases}$

using equation $u_i = \begin{cases} |L_i| & \text{when } |L_i| \le M \\ M & |L_i| > M \end{cases}$

we can modify equation ②.

① =

when $|L_i| \le M$ then $u_i = |L_i|$

So equation ① becomes =

$$\left((|L_i|)^2 + 2M|L_i| - 2M|L_i| \right) = L$$

(when $|L| \le M$)

② =

also when $|L_i| > M$ then $u_i = M$

So equation ② becomes =

$$\left(M^2 - 2M|L_i| - 2M^2 \right) = M(2|L| - M)$$

(when $|L| > M$)

by writing above two cases together we can write =

$$\underset{\theta}{\text{minimize}}\ f(\theta) = \begin{cases} L & |L| \le M \\ M(2|L| - M) & |L| > M \end{cases}$$

(Huber loss function)

which is same as saying =

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{n} h(L_i) \qquad [L_i = |\phi_i^T \theta - \psi_i|]$$

$$\Rightarrow \underset{\theta}{\text{minimize}} \sum_{i=1}^{n} h\left(\phi_i^T \theta - \psi_i\right)$$

Hence both of these equations in part (a) and (b) are equivalent. and soln of (b) can be obtained by solution of (a).

4. (30 points) We test the performance of three regression methods on the wine data set `http://archive.ics.uci.edu/ml/datasets/Wine+Quality`. We will only consider the red wine data set, with 1599 samples. We use the first 1400 samples for training, and the last 199 samples for testing. The goal is to build a linear model of the first 11 features (together with a constant term) to predict the quality of the wine. All models are trained by solving the following optimization problem

$$\underset{\boldsymbol{w},\beta}{\text{minimize}} \quad \sum_{i=1}^{n} \ell(\boldsymbol{x}_i^\top \boldsymbol{w} + \beta - y_i),$$

where the loss functions are

- least squares loss $\ell(t) = t^2$
- Huber loss defined in the previous problem, with $M = 1$
- hinge (deadzone-linear) loss

$$\ell(t) = \begin{cases} 0 & |t| \le 0.5 \\ |t| - 0.5 & |t| > 0.5 \end{cases}$$

The least squares loss can be directly solved by the command `Phi\y` for some properly defined `Phi`. For the latter two, you will use the `cvx` package found on Prof. Boyd's website `https://web.stanford.edu/~boyd/software.html`. Report their prediction performance on the test set using a different metric, mean absolution error (MAE), defined as $(1/n)\sum_{i=1}^{n} |y_i - \hat{y}_i|$.

Ans-4 = I have submitted the working code for this question along with this assignment. Please check the attached files.

(a) = least squares loss $\ell(t) = t^2$

MAE using phi matrix = 0.674940

Prepared phi matrix using various quadratic basis functions for 11 features. Please check code & comments for more info.

(b) = Huber loss function with M=1

MAE using Huber loss = 0.53271
(using CVX package)

(C) = Hinge loss function ⇒

MAE using Hinge loss = 0.54810
(using CVX package)

among these three functions for loss my model is giving least MAE using Huber loss.

$$MAE_{\text{Huber loss}} < MAE_{\text{Hinge loss}} < MAE_{\text{least squares}}$$

Note = Please check code for all phi matrix, CVX loss function Implementation.

(2) = I have implemented least squares using phi matrix & CVX package both but Here I am giving phi-matrix results.

Model training & testing Glossary =

① = available data was devided into training & test data sets. each set had feature vectors data & outcome data seprately (x_train, y_train, x_test, y_test)

② = least squares problem was solved by forming ϕ matrix using quadratic basis functions for features. quadratic function was used b/c it was accurate at the same time it was generalizing.

for $\phi^{\dagger}$ = Pseudo inverse was calculated

$$(\phi^T \phi)^{-1} \phi^T$$

Then $\theta = \phi^{\dagger}(t\_train)$

after getting θ parameters. predictions were made on test data, these Prediction values were fed to Activation function (Sigmoid) to obtain classes.

(3) = Huber loss & Hinge loss both were optimized using CVX packages. and parameters in $(xw + \beta)$ were obtained. (w, β). using these paramet ers prediction on test data was made. these prediction values were fed to activation function (Sigmoid) to obtain classes.

(4) = after making predictions MAE was calculated.

5. (30 points) We test the performance of three classification methods on the ionosphere data set `https://archive.ics.uci.edu/ml/datasets/ionosphere`. There are 351 samples. We use the first 300 samples for training, and the last 51 samples for testing. The goal is to build a linear model of the 34 features (together with a constant term) to predict the binary (± 1) outcome. All models are trained by solving the following optimization problem

$$\underset{\boldsymbol{w},\beta}{\text{minimize}} \quad \sum_{i=1}^{n} \ell(\boldsymbol{x}_i^\top \boldsymbol{w} + \beta, y_i),$$

where the loss functions are

- least squares loss $\ell(t, y) = (t - y)^2$
- logistic loss $\ell(t, y) = \log(1 + \exp(-yt))$
- hinge loss $\ell(t, y) = \max(0, 1 - yt)$

Again, you will use the backslash command to solve for the first model, and `cvx` to solve for the latter two. Report their prediction accuracy on the test set.

Ans-4 = I have Submitted the working code for this question along with this assignment. Please check the attached file.

(1) = least Squares loss ⇒ (using ϕ matrix)

Classifier model Accuracy = 90.19 %

(2) = logistic loss ⇒ (using cvx package)

Classifier model Accuracy = 74.50%

(3) = Hinge loss = (using cvx package)

classifier model accuracy = 74.50%

So

least Square Model Accuracy > logistic loss model accuracy = Hinge loss Model Accuracy

Note: Please check code for all phi matrix, CVX loss function Implementation.

(2) = I have implemented least Squares loss using phi matrix & CVX package both but Here I have shared the results I obtained using only phi- matrix method.

(3) = ideally (Hinge loss, logistic loss) should not give equal accuracy but that's what I obtained even after tuning the model.

Model training & testing Glossary =

① = available data was devided into training & test data sets. each set had feature vectors data & outcome data seprately (x_train, y_train, x_test, y_test)

② = least squares Problem was solved by forming ϕ matrix using quadratic basis functions for features. quadratic function was used b/c it was accurate at the same time it was generalizing.

for $\phi^{\dagger}$ = Pseudo inverse was calculated

$$(\phi^T \phi)^{-1} \phi^T$$

Then $\theta = \phi^{\dagger}(t\_train)$

after getting θ parameters. predictions were made on test data, these prediction values were fed to Activation function (Sigmoid) to obtain classes.

(3) = logistic loss & Hinge loss both were optimized using CVX packages. and parameters in $(xw + \beta)$ were obtained. (w, β). using these paramet ers prediction on test data was made. these prediction values were fed to activation function (Sigmoid) to obtain class.

(4) = Accuracy was calculated by comparing available actual outcomes with predictions.